Characterization of Audio Patch Attacks and Adversarial Training

Ethan Horowitz
Pratt School of Engineering
Duke University
Durham, NC 27708
ethan.horowitz@duke.edu

Alec Liu
Pratt School of Engineering
Duke University
Durham, NC 27708
alec.liu@duke.edu

Alanna Manfredini
Pratt School of Engineering
Duke University
Durham, NC 27708
alanna.manfredini@duke.edu

Abstract

Patch attacks are a common technique used to adversarially affect a neural network. There has been extensive research into whether affixing a patch to an image can cause a neural network to misclassify said image, and, if these patches are incorporated into the training, whether the neural network can become more robust to malicious actors. This work investigates whether attack and defense patterns commonly found in image classification settings are consistent within audio classification settings. Naive networks were shown to be vulnerable to adversarial patch attacks, and they performed significantly better after traditional adversarial training. Additionally, patch performance was shown to vary across frequency divisions, laying groundwork to systematically identify network vulnerabilities.

The code can be accessed at https://github.com/ethaniel0/661AudioAdversary/

1 Introduction

1.1 Adversarial Attack

Deep neural networks are extremely powerful tools for identifying features in images, sound, and language. As these networks spread into various places of everyday life, the possibility of misuse and malpractice has substantially increased - especially in the realm of causing a neural network to misclassify information.

An adversarial attack uses slight perturbations of the input - usually in the form of an image - to maliciously change the output of a neural network. This works because DNNs repeatably learn non-robust features - artifacts in an image that are predictable yet volatile, and are easier to learn than broad overarching patterns [1]. This means that slight alterations can be crafted to look like noise while actually editing these non-robust features, while the changes are imperceptible to human observers. This adversarial noise is often created by extending back-propagation to the input layer of the model, effectively training the image in the same way one would train the network itself.

Preprint. Under review.

Adversarial attacks can take many forms. One prominent form is in the form of a patch. An adversarial patch is a small area of adversarial noise that is pasted or added onto an existing input. A patch can trained in a particular location or placed anywhere within the bounds of a given input. A usual goal when using an adversarial patch attack is to make the patch as small and imperceptible as possible, while still being effective. Within the context of an image, this is usually done by setting a limit on how much a pixel can be changed.

Most studies on adversarial attacks are on their application to images. For example, a small patch stuck to a stop sign can make a self-driving car predict that it sees a speed limit sign [2]. However, not as much research has been dedicated to patch attacks on audio. Some success has been found for attacking sound-based models: Guo et. al found that speech-to-text models can be attacked, with added low-frequency patches used to coerce any input to say phrases like "open the door" or "turn on the lights" [3]. Most speech-to-text models, like Mozilla's, rely on spectrograms to process sound - an image-like representation of sound, plotting the intensity of each frequency over time. This means that speech models are susceptible to patch attacks, similar to image classifiers.

1.2 Adversarial Defense

One main defense against adversarial attacks comes through adversarial training, which makes the model learn to rely more on robust features rather than non-robust ones. Adversarial training allows the model to effectively filter out low-amplitude adversarial patches, making any effective patches more noticeable and thus less useful.

To train against adversarial attacks, a model is trained using training data that contains adversarial patches. This can take the form of adding pre-existing adversarial patches - generated from a different network - randomly to images in the dataset, or of generating patches using the current network during training after each iteration of back-propagation.

Using pre-trained adversarial noise can work for adversarial training because adversarial noise is often transferable. The learned decision boundaries between categories are often similar regardless of network architecture - making adversarial noise from one NN viable for another NN [4]. Thus, training a network to see through these patches can allow it to be nearly as good at defending against adversarial attacks as if the noise had been trained on itself.

2 Related Works

Audio Classification Becker et al. [5] demonstrates the ability to build neural networks for audio classification. In this work, Becker et al. compiled the AudioMNIST database, then used spectrograms to train AlexNet to classify spoken digits.

Patch Attacks Guo et al. [3] demonstrated the effectiveness of targeted patches in larger speech-to-text models. Low-frequency targeted patches were trained for a spectrogram-based speech recognition model, with the goal of creating human-imperceptible attacks. These patch attacks succeeded, and were able to coerce the network's output into the desired text most of the time. These patches were universal - instead of a command-based attack, the patches can be trained to elicit any targeted response. They were also shown to be viable in real-world scenarios, and could be administered in real life or over the air.

Adversarial Training Bai et al. [6] reviewed various adversarial training techniques on robustness. From prior research, it was found that adversarial training significantly increases robustness for networks trained on datasets with few categories, such as MNIST or CIFAR-10. For datasets with more categories, simple adversarial training often only allows for accuracies of 45-50% with adversarial training. Other strategies were reviewed: the use of strategies like curriculum-based learning - where the network is trained on patches with increasingly many PGD iterations, ensemble training, adaptive epsilon values, and replacing the expensive PGD with Free-AT - where gradients are reused. Each of these training methods were shown to increase robustness over classic PGD or FGSM training.

Additionally, Madry et al. [7] found that Projected Gradient Descent (PGD) is by far the strongest first-order attack. This means that it is effective for attacking images with strongly localized information.

In fact, the authors posit that adversarial training is usually used as the baseline of defense because of how consistently well it performs.

3 Methodology

3.1 Dataset

This research was performed with a selection of 30,000 data points from the AudioMNIST dataset, split into 70% train, 10% validation, and 20% test. This database consists of .wav files of the digits 0-9 spoken by 60 people of various genders and accents [5].

3.2 Training the Base Model

To prepare the data for initial training, each .wav file was resampled at 11,000 samples/second and padded to a length of 16,000. Then, each waveform was converted to a Mel frequency spectrogram with 256 Mel frequencies across 32 time windows. Finally, each spectrogram was converted to a tensor, allowing it to be treated as an image-like input by the network.

A derivation of the AlexNet model was created with five convolutional and ReLU layers, with pooling after the first and second layer; and three fully connected layers with ReLU and dropout after layers 6 and 7. This network was then trained on the tensors to identify digits [5].

3.3 Patch Generation and Defense

3.3.1 Patch Training

A set of additive adversarial patches were trained to disrupt the accuracy of the AlexNet model at classifying spectrograms. These were trained by creating a mask for the spectrograms with a small patch section that was not zeroed out. PGD was performed for each patch, and the effectiveness of the patch against the various spoken digits was determined. To experiment with patches, 279 positioned patches were created: for the 32x256 image sizes, the shapes of (32x1), (32x2), (32x4), (1x256), (2x256), (4x256), (4x4), (6x6), and (8x8) were used, and each shape was positioned every 10 pixels vertically and horizontally. For un-positioned patches, the same shapes were tested, as well as (4x8), (4x16), (8x4), and (16x4). These patches fell into three main categories: horizontal bars, spanning all timeframes; vertical bars, spanning all frequencies; and squares / rectangles. The effectiveness and perceptibility of each type of patch were compared.

This process was repeated with targeted patches. When training targeted patches, all patches described above were trained for every number - that is, all 279 positioned patches were trained for the number 0, 1, 2, etc., and all 13 un-positioned patch shapes were trained for each number as well.

3.3.2 Adversarial Training

A second AlexNet model was trained on both the unadulterated spectrograms and the spectrograms with the untargeted patches overlaid on the images. Then, more patches were created, and all models were tested on the accuracy of their predictions against clean data, data with patches they had been trained with, data with patches in the same position as those they had been trained with, data with global patches, and data with targeted patches.

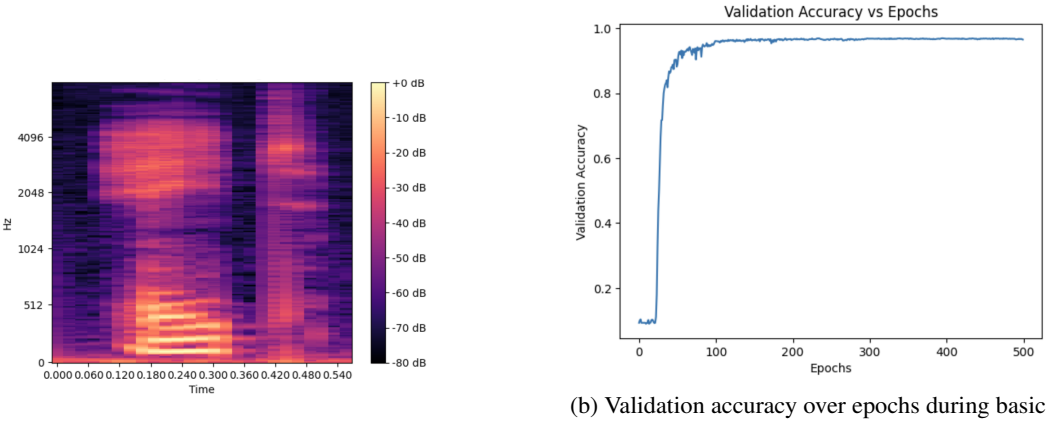
4 Experimental Results

4.1 Data Preparation and Training

To prepare the dataset, each .wav file was transformed as described in Methodology. An example spectrogram for the digit 8 is shown in (Figure 1a).

Model training was performed for 500 epochs with a batch size of 100. The additional epochs after the model accuracy became saturated were executed to show that the model stabilized in accuracy after 150 epochs and highlighted that no additional hidden features could be learned with extra

training, nor would the model overfit with extended training. The training accuracy was low until epoch 30, where there were some great jumps in accuracy (Figure 1b).



(a) Example Mel spectrogram for the digit 8

(b) Validation accuracy over epochs during basic training

Figure 1: Training input data and progress

4.2 Adversarial Attack

Over 3000 patches were produced of various sizes and shapes and trained with PGD. Local patches were trained in a fixed location in the image, while global patches were trained in multiple randomly-generated locations in images. The best ten of these two groups of untargeted patches are depicted in Figure 2 alongside their shape, location (if local) and the resulting accuracy of the naive model. Patches were trained to be constrained to various parts of the spectrogram. The confusion matrices after applying the best patch from each location are depicted in Figure 3 on the following page. Additionally, targeted patches were trained to convince the model each spectrogram was a specific number from 0 to 9.

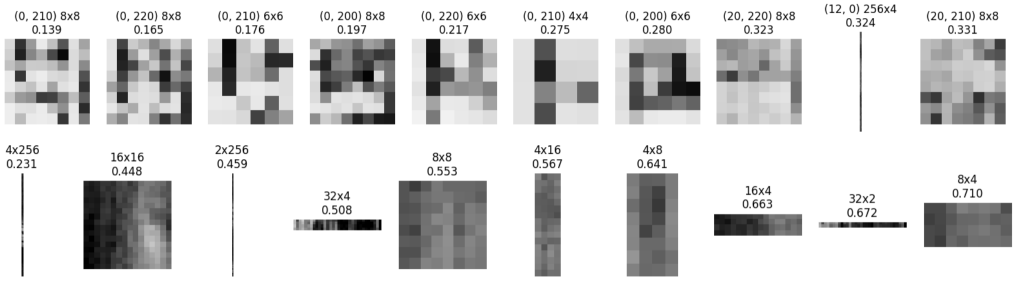


Figure 2: Most effective local (top) and global (bottom) patches, with position, size, and accuracy

4.3 Adversarial Defense

The same model architecture was adversarially trained with a set of untargeted local patches augmented within the dataset. The performance of the robust and naive models were compared across a variety of categories: clean test data, test data augmented with the untargeted training patches, test data augmented with new patches with the same shape and location as the untargeted training patches (local), test data augmented with new patches with a different shape and location as the untargeted patches (global), and test data augmented with new targeted patches. Results are summarized in Table 1 below. The fields corresponding to the naive network's performance on known untargeted patches and unknown untargeted global patches are not applicable, since the naive network was not trained on any untargeted patches.

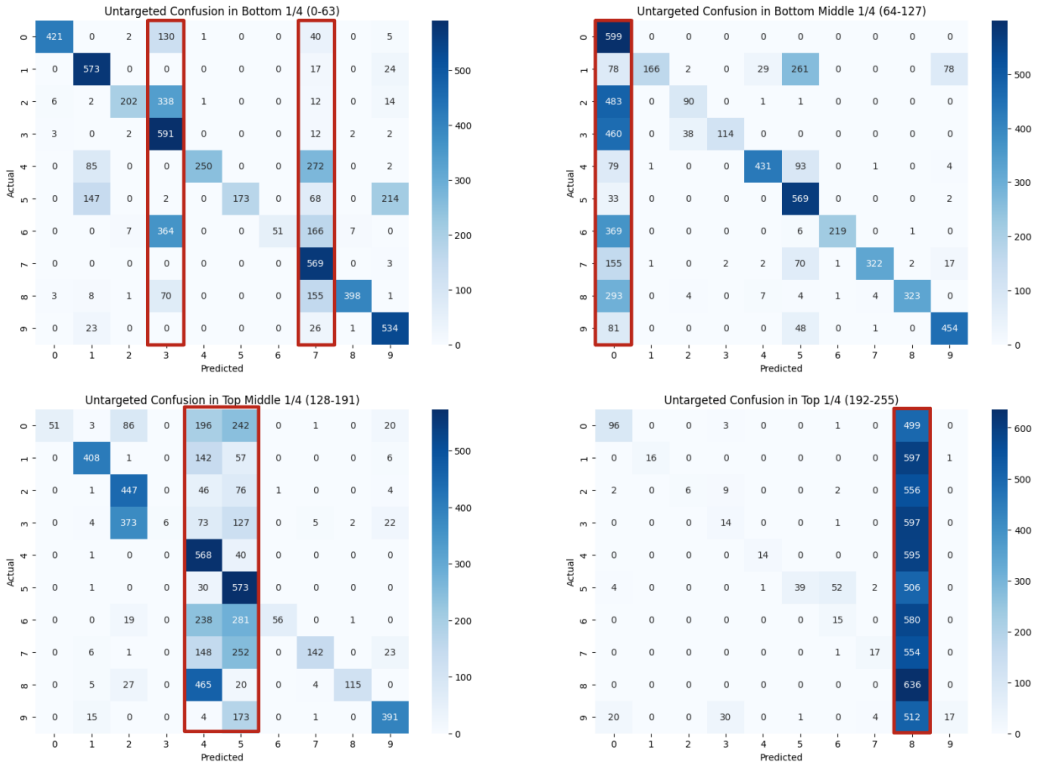


Figure 3: Confusion matrices across frequency quartiles

Table 1: Testing accuracy by patch type and training type

		Attacking				
		Clean	Known Untargeted	Unknown Untargeted (Local)	Unknown Untargeted (Global)	Unknown Targeted
Training	None	97.24%	N/A	24.02%	N/A	30.65%
	Untargeted (Local)	95.41%	95.01%	93.31%	88.80%	39.58%

5 Discussion

5.1 Adversarial Attack

The adversarial attack produced results which had interesting implications for the shape and location of the patches in future experiments. The most important takeaway was that the most effective local, untargeted patches had certain patterns within them, which the targeted and global patches did not have.

Almost all of the untargeted, local patches were at the start. 70% of the most successful patches were located at time t = 0. Also, as can be seen in Figure 2, the first, second and third columns of the patches located near (0, 210) have a similar pattern of intensity irrespective of size; the first column usually has higher intensity noise, the second has lower noise and the third has higher intensity noise again. Finally, the local, untargeted results are highly dependent on frequency range. In Figure 3, all the patches were filtered with respect to their frequency range and split into groups to be analyzed. In the figure, this was done in the range of the bottom quarter, the bottom half, the bottom three quarters and all frequencies. The results of this experiment highlighted that when the patches were given

the freedom to be in any frequency range, the untargeted patches acted as pseudo-targeted patches: despite being trained to merely disrupt the classification, the best untargeted patch heavily targeted the digit 8. On the other hand, when patches were restricted to be in only the lower three-quarters of the frequencies, most of the numbers were misclassified as a four or five. This study was also run including only high frequencies with similar results. In the future, it would be interesting to run this experiment with restrictions on the time frames the patches can affect. By dividing up the spectrogram in this way, this method acts similarly to an attention mechanism by highlighting the frequencies which have the most impact on correct classification by our model.

On the other hand, by their nature, global patches are not dependent on frequency and thus become highly dependent on area. In Figure 2, it can be seen that the global patches with larger areas tend to have more success.

The targeted patches that most effectively reduced network accuracy - without considering if they actually confused the network into misclassifying the digit to the desired number - were almost all of shape 256x4. The one exception was the targeted patch for the digit 8 (Figure 4, Appendix A). This makes sense, as 256x4 is a large patch. Also, since the targeted 8-patch does not follow the same pattern, this further reemphasises that the untargeted patches effectively became targeted patches for the digit 8: many digits can be easily misclassified as an 8, and it does not need a large patch to do so.

Finally, many of the best performing patches were compared aurally. This showed that although the most imperceptible patches were not as effective as other patches at disrupting the model, they still dropped the accuracy to approximately 60%. More concerning, however, were the patches that spanned a large range of frequency and few timeframes. These patches sound similar to someone hitting their microphone, or an audio glitch, but decreased model accuracy to approximately 23%.

5.2 Adversarial Defense

Compared to adversarial training in image classification networks, similar trends were observed for the naive and robust audio networks. First, the naive network showed poor performance on both untargeted and targeted attacks with 24.02% and 30.65% test accuracy respectively, confirming that the model naturally learned non-robust features within the dataset. Second, the robust model showed a slight decrease in accuracy on the clean dataset, confirming the tradeoff between accuracy and robustness typically seen during adversarial training. Third, the model showed high performance against the training patches, even with patches being applied to different samples. Additionally, the model showed significantly higher accuracy against new types of patches, demonstrating that the model had learned robust features. Impressively, the model still modestly improved against targeted patch attacks, despite never encountering them during training.

6 Conclusion

This work confirms that many attack techniques used for image classification also transfer well to the audio classification domain. The effectiveness and imperceptibility observed among many adversarial patches highlights the importance of fostering robust learning in speech applications. Additionally, this work confirms that adversarial training is an efficacious defense technique. Although only exposed to a few patches during training, the network showed improved performance against a variety of adversarial patches; future research should investigate the effect of increasing the variation between training patches on overall model robustness. Finally, patch effectiveness was shown to vary across frequency divisions; future research should leverage these differences to develop techniques to systematically identify network vulnerabilities.

References

[1] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," arXiv.org, Aug. 12, 2019. https://arxiv.org/abs/1905.02175

[2] I. Evtimov et al., "Robust Physical-World Attacks on Deep Learning Models," arXiv (Cornell University), Jan. 2017, doi: https://doi.org/10.48550/arxiv.1707.08945.

[3] Guo, Hanqing, et al. "Specpatch." Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 7 Nov. 2022, https://doi.org/10.1145/3548606.3560660.

[4] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," arXiv (Cornell University), Jan. 2016, doi: https://doi.org/10.48550/arxiv.1611.02770.

[5] S. Becker et al., "AudioMNIST: Exploring Explainable Artificial Intelligence for audio analysis on a simple benchmark," Journal of the Franklin Institute, vol. 361, no. 1, pp. 418–428, Jan. 2024. doi:10.1016/j.jfranklin.2023.11.038

[6] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent Advances in Adversarial Training for Adversarial Robustness," arXiv (Cornell University), Feb. 2021, doi: https://doi.org/10.48550/arxiv.2102.01356.

[7] A. Madry, Aleksandar Makelov, L. Schmidt, Dimitris Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," no. 4, Jun. 2017, doi: https://doi.org/10.48550/arxiv.1706.06083.

A Targeted Patch Performance Matrices

Figure 4: Best local targeted patches, organised by test accuracy

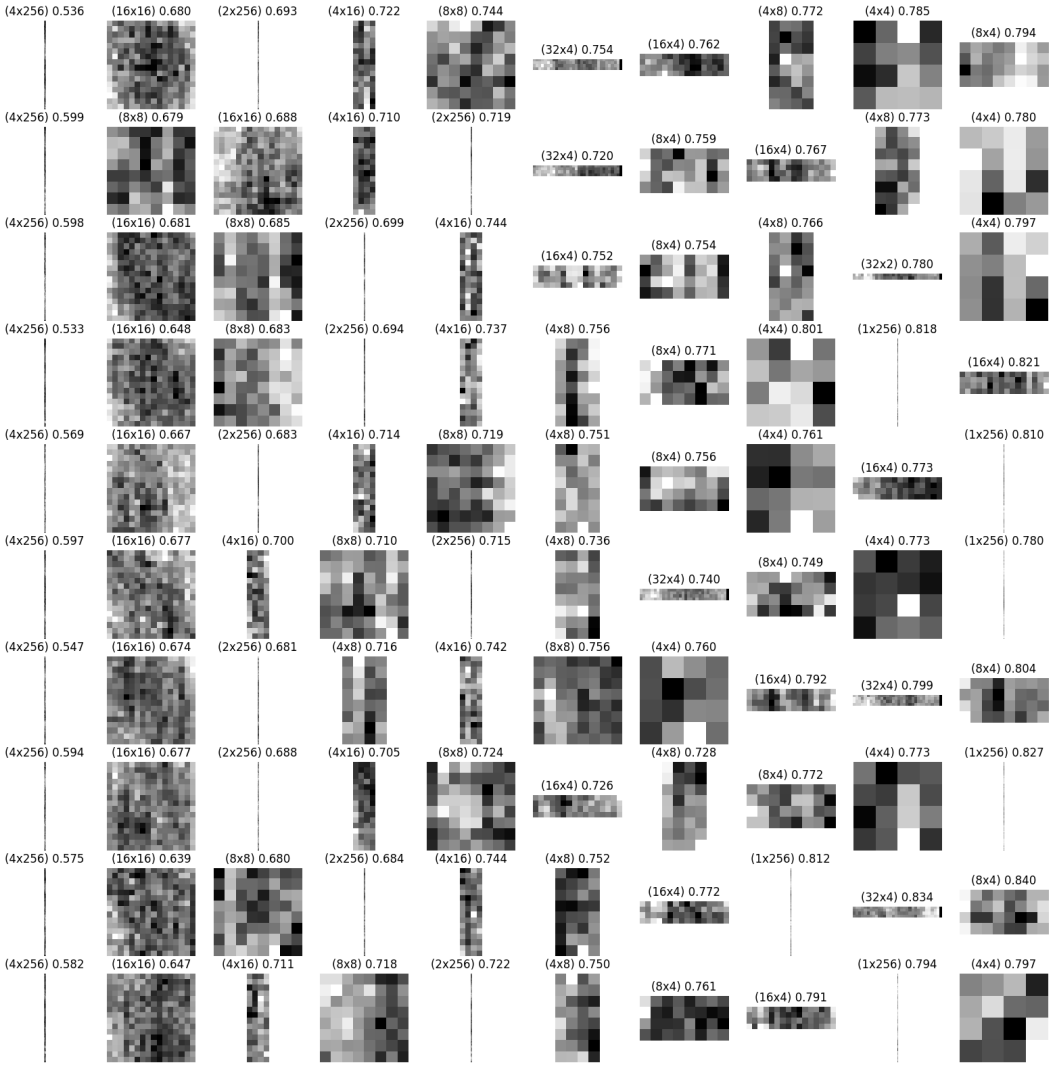


Figure 5: Best global targeted patches, organised by test accuracy

Figure 6: Best local targeted patches, organised by number of resulted correctly-targeted outputs

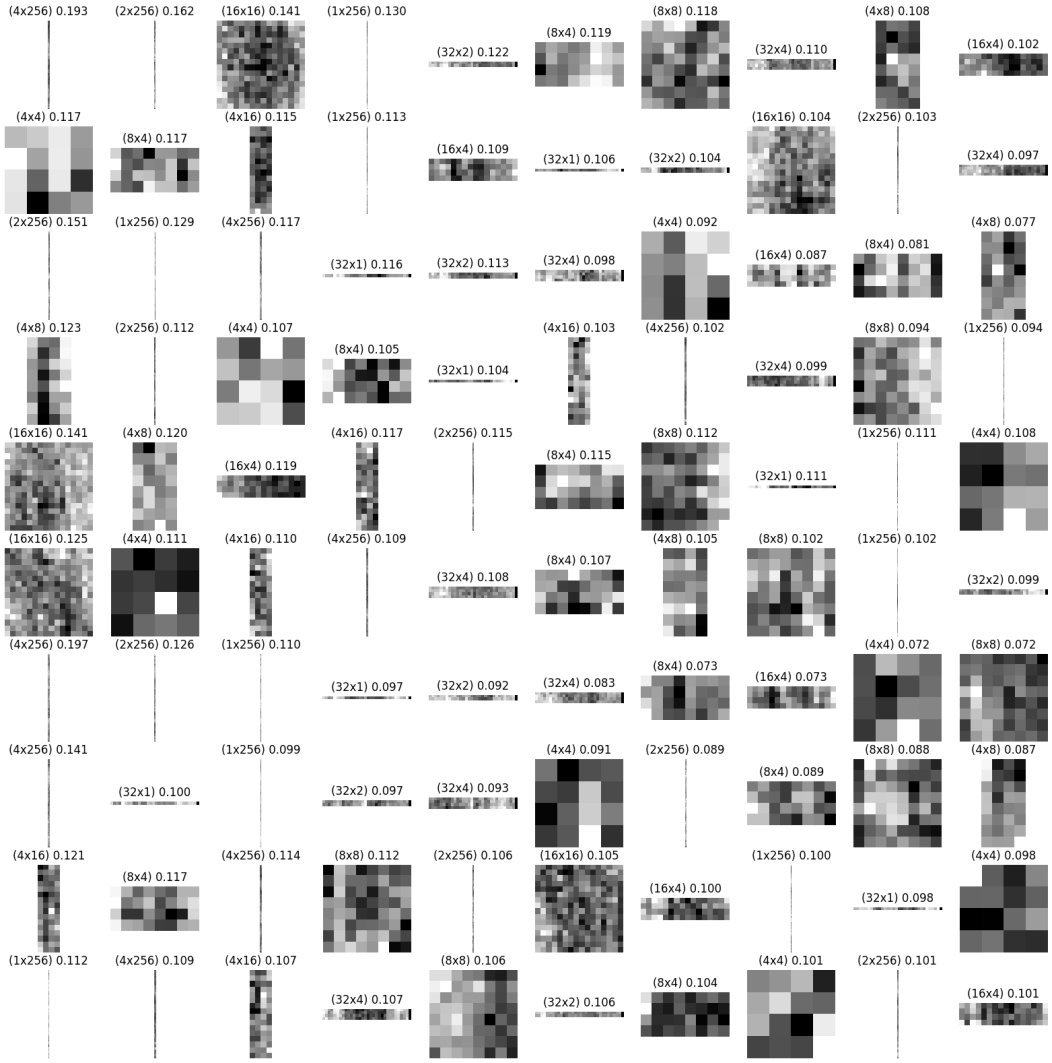


Figure 7: Best global targeted patches, organised by number of resulted correctly-targeted outputs